Sistema de Gestión de Cadena de Deportes - Documentación Técnica

Bibliotecas y Estructura Base

El sistema está construido sobre un conjunto esencial de bibliotecas de C++ que proporcionan las funcionalidades necesarias para una aplicación de gestión comercial robusta:

La biblioteca iostream proporciona la base para la interacción con el usuario a través de una interfaz de consola clara y amigable. Esta interacción es fundamental en nuestro sistema, ya que necesitamos constantemente recibir y validar datos del usuario.

La biblioteca pqxx/pqxx es el pilar de nuestra conexión con PostgreSQL. Esta biblioteca nos permite implementar un manejo seguro de transacciones y consultas parametrizadas, fundamentales para mantener la integridad de los datos en un sistema comercial.

Para complementar estas funcionalidades básicas, el sistema incorpora varias bibliotecas estándar de C++:

- string y sstream: Para el procesamiento y validación de texto

- limits: Para establecer límites en la entrada de datos

- iomanip: Para formatear la presentación de precios y datos numéricos

- set: Para gestionar colecciones únicas, como los horarios registrados

- algorithm: Para implementar validaciones eficientes

Funciones de Utilidad y Validación

El sistema implementa un conjunto robusto de funciones auxiliares que podemos agrupar en tres categorías principales:

Funciones de Interfaz de Usuario:

La función limpiarPantalla() mantiene una interfaz ordenada y profesional, mientras que pausar() permite al usuario leer y procesar la información antes de continuar. Estas funciones son esenciales para crear una experiencia de usuario fluida.

Funciones de Validación:

El sistema incluye validaciones exhaustivas para cada tipo de dato:

validarDNI(): Asegura que los documentos de identidad:

- Contengan exactamente 11 dígitos numéricos

- No incluyan caracteres especiales o letras

validarCUIT(): Similar a validarDNI(), pero específico para identificadores de empresas:

- Verifica la longitud exacta de 11 dígitos

- Asegura que solo contenga números

validarHora(): Implementa una validación completa de horarios:

- Verifica el formato HH:MM

- Valida rangos de horas (0-23) y minutos (0-59)

- Asegura la presencia del separador ':'

validarMonto(): Garantiza la validez de valores monetarios:

- Acepta solo números positivos

- Permite valores decimales

- Previene entradas no numéricas

Estructura de la Base de Datos

El modelo de datos implementa una estructura normalizada que representa el negocio de una cadena de deportes:

Sucursales:

- ID único autogenerado

- Domicilio y ciudad

- Relaciones con empleados y productos

Empleados:

- Legajo autogenerado

- Datos personales completos (nombre, DNI, domicilio)

- Teléfonos de contacto (relación uno a muchos)

- Horarios por sucursal (relación muchos a muchos con datos adicionales)

Fábricas:

- CUIT único

- Información comercial completa

- Restricción de un único producto por fábrica

Productos:

- Código autogenerado

- Descripción y características

- Relación única con fábrica fabricante

- Precios variables por sucursal

Normalización

Primera Forma Normal (1NF):

El diseño cumple con 1NF mediante:

- Claves primarias únicas en todas las tablas

- Teléfonos de empleados normalizados en tabla separada

- Horarios de trabajo en tabla independiente

- Precios por sucursal en tabla específica

Segunda Forma Normal (2NF):

Se logra mediante:

- Dependencias completas de claves primarias

- Tablas intermedias para relaciones N:M (EmpleadosSucursales)

- Separación de datos de productos y fábricas

Tercera Forma Normal (3NF):

Se implementa a través de:

- Eliminación de dependencias transitivas

- Normalización de datos de ubicación

- Separación de precios por sucursal

Funciones Principales

Las funciones principales del sistema implementan un manejo cuidadoso de transacciones:

registrarEmpleado():

Esta función demuestra un manejo completo de datos relacionales:

- Validación previa de existencia de sucursales

- Transacción única que abarca múltiples inserciones

- Manejo de colecciones de teléfonos y horarios

- Validaciones en cascada de datos relacionados

registrarProducto():

Implementa la lógica de negocio para productos y fábricas:

- Verifica la restricción de producto único por fábrica

- Maneja precios variables por sucursal

- Valida existencia de sucursales antes de asignar precios

- Mantiene la integridad referencial

consultarProductosSucursal() y consultarHorariosEmpleados():

Implementan consultas complejas que:

- Utilizan joins optimizados

- Manejan agrupaciones eficientes

- Presentan resultados formateados

- Incluyen totales y resúmenes

Análisis de Consultas SQL

Las consultas SQL implementadas en el sistema demuestran un diseño cuidadoso:

Consultas de Verificación:

Para validar la existencia de sucursales:

SELECT 1 FROM Sucursales LIMIT 1

Esta consulta simple pero efectiva se utiliza para verificar precondiciones antes de operaciones complejas.

Consultas de Horarios:

SELECT e.legajo, e.nombre, e.dni, h.hora\_inicio, h.hora\_fin

FROM HorariosEmpleados h

JOIN Empleados e ON h.legajo = e.legajo

WHERE h.sucursal\_id = $1 AND h.dia\_semana = $2

ORDER BY h.hora\_inicio

Esta consulta demuestra:

- Joins optimizados para relacionar datos

- Ordenamiento significativo para el usuario

- Parámetros para prevenir inyección SQL

Consultas de Productos:

SELECT p.codigo, p.descripcion, p.color, p.costo\_fabricacion,

ps.precio\_venta, f.nombre as fabricante, f.pais\_origen

FROM ProductosSucursales ps

JOIN Productos p ON ps.producto\_codigo = p.codigo

JOIN Fabricas f ON p.fabrica\_cuit = f.cuit

WHERE ps.sucursal\_id = $1

ORDER BY p.descripcion

Características destacables:

- Múltiples joins para datos relacionados

- Alias descriptivos

- Ordenamiento por campo significativo

Manejo de Transacciones

El sistema implementa un manejo seguro de transacciones mediante bloques try-catch:

cpp

try {

work W(conn);

// Operaciones de base de datos

W.commit();

} catch (const exception &e) {

// Manejo de errores y rollback implícito

}

Este patrón asegura:

- Atomicidad de operaciones complejas

- Rollback automático en caso de error

- Liberación adecuada de recursos

- Manejo consistente de errores

Interfaz de Usuario

La interfaz implementa un diseño consistente usando caracteres ASCII para crear marcos visuales:

\*========================================\*

| GESTIÓN DE EMPLEADOS |

\*========================================\*

Características de la interfaz:

- Menús jerárquicos claros

- Retroalimentación inmediata sobre errores

- Formato consistente para datos monetarios

- Validación en tiempo real de entradas

Consideraciones de Seguridad

El sistema implementa varias capas de seguridad:

- Validación exhaustiva de entrada de usuario

- Consultas parametrizadas para prevenir SQL injection

- Manejo de transacciones para mantener integridad

- Verificaciones de existencia antes de operaciones

- Restricciones a nivel de base de datos

Esta implementación proporciona una base sólida para la gestión de una cadena de deportes, con énfasis en la integridad de datos y la experiencia del usuario.